(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
22.10.2003 Bulletin 2003/43

(51) Int Cl.7: **H03M 13/29**

(21) Application number: **00116858.2**

(22) Date of filing: **04.08.2000**

(54) **Linearly approximated log map algorithm for turbo decoding**

Lineare Approximation des LOG-MAP Algorithmus für Turbodekodierung

Approximation linéaire de l'algorithme LOG-MAP pour turbo décodage

(84) Designated Contracting States:
DE FR GB

(43) Date of publication of application:
20.03.2002 Bulletin 2002/12

(73) Proprietor: **MOTOROLA INC.**
Schaumburg, IL 60196-3750 (US)

(72) Inventors:
• **Sivan, Noam**
55900 Ganey-tikva (IL)
• **Aidan, Fabrice**
43762 Ra'anana (IL)
• **Leshin, Gregory**
78701 Ashkelon (IL)

(74) Representative: **Gibson, Sarah Jane et al**
Motorola
European Intellectual Property Section
Law Department
Midpoint
Alencon Link
Basingstoke, Hampshire RG21 7PL (GB)

(56) References cited:
US-A- 5 933 462

• YUNG-FU CHEG, TONY OTTOSSON: "Linearly Approximated Log-MAP Algorithms for Turbo Decoding" 51ST IEEE VEHICULAR TECHNOLOGY CONFERENCE PROCEEDINGS, vol. 3, 15 - 18 May 2000, pages 2252-2256, XP002156559 Tokyo
• "SIMPLIFIED LOG-MAP ALGORITHM" RESEARCH DISCLOSURE,INDUSTRIAL OPPORTUNITIES LTD. HAVANT,GB, no. 421, May 1999 (1999-05), page 612 XP000888685 ISSN: 0374-4353
• ROBERTSON P ET AL: "A COMPARISON OF OPTIMAL AND SUB-OPTIMAL MAP DECODING ALGORITHMS OPERATING IN THE LOG DOMAIN" PROCEEDINGS OF THE CONFERENCE ON COMMUNICATIONS (ICC),US,NEW YORK, IEEE, 18 June 1995 (1995-06-18), pages 1009-1013, XP000533149 ISBN: 0-7803-2487-0

EP 1 189 355 B1

Description

FIELD OF THE INVENTION

[0001] Apparatus and method for implementing a linearly approximated Log MAP algorithm and especially an apparatus and method for implementing a linearly approximated Log MAP algorithm for turbo decoding and turbo equalization.

BACKGROUND OF THE INVENTION

[0002] Turbo Coding (i.e.- TC) is used for error control coding in digital communications and signal processing. The following references give some examples of various implementations of the TC: "Near Shannon limit error correcting coding and decoding: turbo-codes", by Berrou, Glavieux, Thitimajshima, IEEE International Conference of Communication. Geneva Switzerland, pp. 1064-1070, May 1993; "Implementation and Performance of a Turbo/MAP Decoder", Pietrobon, International Journal of Satellite Communication; "Turbo Coding", Heegard and Wicker, Kluwer Academic Publishers 1999.

[0003] MAP algorithm and soft output Viterbi algorithm (SOVA) are Soft Input Soft Output (i.e.- SISO) decoding algorithms that have gained wide acceptance in the area of communications. Both algorithms are mentioned in U.S patent 5,933,462 of Viterbi et al.

[0004] The TC has gained wide acceptance in the area of communications, such as in cellular networks, modems, and satellite communications. Some turbo encoders consists of two parallel-concatenated systematic convolutional encoders separated by a random interleaver. A turbo decoder has two soft-in soft-out (SISO) decoders. The output of the first SISO is coupled to the input of the second SISO via a first interleaver, while the output of the second SISO is coupled to an input of the first SISO via a feedback loop that includes a deinterleaver.

[0005] A common SISO decoder uses either a maximum a posteriori (i.e.- MAP) decoding algorithm or a Log MAP decoding algorithm. The latter algorithm is analogues to the former algorithm but is performed in the logarithmic domain. Another common decoding algorithm is the max log MAP algorithm. The max log MAP is analogues to the log MAP but the implementation of the former involves an addition of correction factor. Briefly, the MAP finds the most likely information bit to have been transmitted in a coded sequence.

[0006] The output signals of a convolutional encoder are transmitted via a channel and are received by a receiver that has a turbo decoder. The channel usually adds noise to the transmitted signal.

[0007] During the decoding process a trellis of the possible states of the coding is defined. The trellis includes a plurality of nodes (states), organized in T stages, each stage has $N=2^{K-1}$ nodes, whereas T be-

ing the number of received samples taken into account for evaluating which bit was transmitted from a transmitter having the convolutional encoder and K is the constraint length of the code used for encoding. Each stage is comprised of states that represent a given time. Each state is characterized by a forward state metric, commonly referred to as alpha (α or a) and by a backward state metric, commonly referred to as beta (β or b) . Each transition from a state to another state is characterized by a branch metric, commonly referred to as gamma (γ).

[0008] Alphas, betas and gammas are used to evaluate a probability factor that indicates which signal was transmitted. This probability factor is commonly known as lambda (Λ). A transition from a stage to an adjacent stage is represented by a single lambda.

[0009] A function $MAX^*(a(n),b(n))$ is frequently used when alphas, betas and gammas are calculated. Conveniently, said calculation involves in a comparison of a (n) and b(n). Said elements a(n) and b(n) usually have real values. $MAX^*(a(n), b(n))$ equals $MAX(a(n), b(n))$ + $Log(1+EXP\{-|a(n)-b(n)|\})$. The first portion of said equation is usually calculated. The calculation of the second portion is relatively complicated and time consuming. Usually an approximation of said second portion is calculated. Said approximation is either a linear approximation, a step approximation or a multi step approximation. A linear approximation method is described in "Linearity Approximated for Log-MAP Algorithms for Turbo Decoding", by Jung-Fu Cheng and Tony Ottoson, 51st IEEE Vehicular Technology Conference Proceedings, 13-15 May 2000, pages 2252-2256. Said method suggests to approximate the second portion by a the following functions: $MAX\{0, (C-|a(n)-b(n)|/4\}$ and by : $MAX\{0, (C-|a(n)-b(n)|/8\}$. Both estimations provide relatively poor performances when |a(n)-b(n)| is relatively small, and their implementation is relatively time consuming.

[0010] Another simplified linear approximation method is described in "Simplified Log-MAP Algorithm", Research Disclosure, No. 421, XP888685, May 1999, page 612. There is a need for providing a fast and high performance apparatus and method for implementing a linearly approximated Log MAP algorithm.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] While the invention is pointed out with particularity in the appended claims, other features of the invention are disclosed by the following detailed description taken in conjunction with the accompanying drawings, in which:

FIG. 1. is a schematic description of a portion of a $MAX^*(a(n),b(n))$ function and a linear approximation of said function, according to a preferred embodiment of the invention;

FIGS. 2-3 are schematic diagrams illustrating two apparatuses for implementing a linearly approxi-

mated Log MAP algorithm, according to a preferred embodiment of the invention; and

FIG. 4 is a flow chart illustrating a method for implementing a linearly approximated Log MAP algorithm, according to a preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE DRAWINGS

[0012] It should be noted that the particular terms and expressions employed and the particular structural and operational details disclosed in the detailed description and accompanying drawings are for illustrative purposes only and are not intended to in any way limit the scope of the invention as described in the appended claims.

[0013] The invention provides a fast and high performance apparatus and method for implementing a linearly approximated Log MAP algorithm. Said approximation is a linear approximation having a slope of -0.5, said slope providing an improved approximation when $a(n)$ is substantially equal to $b(n)$. Conveniently, method is executed during a single clock cycle.

[0014] The invention provides a fast and high performance apparatus and method for implementing a linearly approximated Log MAP algorithm.

[0015] The invention provides a method for implementing a linearly approximated Log MAP algorithm, said implementation involves calculating MAX*(a(n),b(n)) function, said method comprising the steps of: (A) Receiving $a(n)$, $b(n)$ and DE; (B) Calculating $(a(n)+b(n)+DE)/2$ and generating at least one intermediate result, said at least one intermediate result reflecting at least one relationship between at least two elements out of a(n), b(n) and DE; and (C) Providing an MAX*(a(n),b(n)) result selected from a group comprising of a(n), b(n) or (a(n)+b(n)+DE)/2, said selection dependent upon the at least one intermediate result. Conveniently, the linearly approximated Log MAP algorithm is implemented in either an iterative decoding process or in an iterative equalization process.

[0016] The invention provides a method for implementing a linearly approximated Log MAP algorithm wherein a first intermediate result indicates whether an absolute value of a difference between a(n) and b(n) is greater than DE. A second intermediate result indicates whether a(n) is greater than b(n). Conveniently, the at least one intermediate result comprising of either a sign(a(n)-b(n)-DE) signal and a sign(b(n)-a(n)-DE) signal or of a sign(a(n)-b(n)) signal and a sign(|a(n)-b(n)|-DE) signal.

[0017] The invention provides a method for implementing a linearly approximated Log MAP algorithm wherein the MAX*(a(n),b(n)) result equals (I) MAX(a(n), b(n)) when |a(n)-b(n)| is not smaller than DE, and (II) (a(n)+b(n)+DE)/2 when |a(n)-b(n)| is smaller than DE. |a(n)-b(n)| is smaller than DE when (a(n)-b(n)-DE) < 0 and (-a(n)+b(n)-DE) < 0.

[0018] The invention provides a method for implementing a linearly approximated Log MAP algorithm wherein the MAX*(a(n),b(n)) result equals (I) a(n) when (a(n)-b(n)-DE) > 0 and (b(n)-a(n) -DE) < 0; (II) b(n) when (a(n)-b(n)-DE) < 0 and (b(n)-a(n) -DE) > 0; and (III) (a(n)+b(n)+DE)/2 when |a(n)-b(n)| is smaller than DE.

[0019] The invention provides an apparatus for implementing a linearly approximated Log MAP algorithm, said implementation involves calculating MAX*(a(n),b(n)) function, said apparatus comprising of: (A) Calculating means for receiving a(n), b(n) and DE; for Calculating (a(n)+b(n)+DE)/2 and for generating at least one intermediate result, said at least one intermediate result reflecting at least one relationship between at least two elements out of a(n), b(n) and DE; and (B) Selecting means for providing an MAX*(a(n),b(n)) result selected from a group comprising of a(n), b(n) or (a(n)+b(n)+DE)/2, said selection dependent upon the at least one intermediate result. Conveniently, the linearly approximated Log MAP algorithm is implemented in an iterative decoding process or in an iterative equalization process.

[0020] The invention provides an apparatus for implementing a linearly approximated Log MAP algorithm wherein a first intermediate result indicates whether an absolute value of a difference between a(n) and b(n) is greater than DE and a second intermediate result indicates whether a(n) is greater than b(n). Conveniently, the at least one intermediate result comprising of a sign(a(n)-b(n)-DE) signal and a sign(b(n)-a(n)) signal. The at least one intermediate result comprising of a sign(a(n)-b(n)) signal and a sign(|a(n)-b(n)|-DE) signal.

[0021] The invention provides an apparatus for implementing a linearly approximated Log MAP algorithm that is adapted to calculate the MAX*(a(n),b(n)) function during a single clock cycle.

[0022] The invention provides an apparatus for implementing a linearly approximated Log MAP algorithm wherein the MAX*(a(n),b(n)) result equals MAX(a(n),b(n)) when |a(n)-b(n) | is not smaller than DE, and equals (a(n)+b(n)+DE)/2 when |a(n)-b(n)| is smaller than DE.

[0023] The invention provides an apparatus for implementing a linearly approximated Log MAP algorithm wherein the MAX*(a(n), b(n)) result equals a(n) when (a(n)-b(n)-DE) > 0 and (b(n)-a(n) -DE) < 0; wherein the MAX*(a(n), b(n)) result equals b(n) when (a(n)-b(n)-DE) < 0 and (b(n)-a(n) -DE) > 0; and wherein the MAX*(a(n), b(n)) result equals (a(n)+b(n)+DE)/2 when |a(n)-b(n)| is smaller than DE.

[0024] For convenience of explanation, the following description relates to turbo decoding. The invention is not limited to said decoding method. For example said invention is also applied in various detection processes such as equalization processes.

[0025] |a(n)-b(n)| is an absolute value of a difference between a(n) and b(n).

[0026] EXP{-|a(n)-b(n)|} equals e by the power of -|a(n)-b(n)|.

[0027] FIG. 1 is a graph illustrating a portion of a MAX* (a(n), b(n)) versus an absolute value of a difference between a(n) and b(n). Said graph further illustrates a linear approximation AP(n) of said portion. Said portion equals Log(1+EXP{-|a(n)-b(n)|}). Curved line 4 illustrates portion Log(1+EXP{-|a(n)-b(n)|}), while line 5 illustrates a linear approximation of said portion. The slope of said linear approximation substantially equals - 0.5. AP(n) = MAX{0,(DE-(a(n)-b(n))/2)}. Preferably DE=2*LOG(2).

[0028] Said linear approximation provides superior performances when the difference between a(n) and b(n) is very small. Furthermore, such an approximation allows to speed the calculation of the MAX* function, as further illustrated in Figs. 2-3.

[0029] Referring to Fig. 2 illustrating an apparatus 10 for implementing a linearly approximated Log MAP algorithm, according to a preferred embodiment of the invention.

[0030] Apparatus 10 implements the following equations :

$$[1] \qquad MAX^*(a(n), b(n)) = MAX(a(n),b(n))$$

when |a(n)-b(n)| is not smaller than DE.

$$[2] \qquad MAX^*(a(n), b(n)) = (a(n)+b(n)+DE)/2$$

when |a(n)-b(n)| is smaller than DE.

[0031] Apparatus 10 allows to calculate in parallel the following values: MAX(A(n), b(n)); (a(n)+b(n)+DE)/2 and the sign of (|a(n)-b(n)| -DE) and then to determine, according to said sign) which of the first two values to provide as a result of the calculation.

[0032] Accordingly, apparatus 10 comprises of first means 11 for calculating the following values: MAX(A(n), b(n)); (a(n)+b(n)+DE)/2 and the sign of (|a(n)-b(n)| -DE) and of a second means 19 to determine which of the first two elements to provide as a result, according to said sign.

[0033] First means 11 comprising of first to third subtracting units 12-14, adder 15 and two multiplexers 16 and 17. First and second subtracting units 12 - 13 and adder 15 receive a(n), b(n) and DE. Third subtracting unit 14 receives a(n) and b(n) and generates sign(a(n)-b(n)). Sign(a(n)-b(n)) is provided a a control signal to first and second multiplexers 16 and 17. Second multiplexer 17 receives a(n) and b(n) and selects the greater element out of them, in response to sign(a(n)-b(n)). Therefore, second multiplexer 17 generates MAX(a(n), b(n)) and provide it to second means 19. Adder 15 receives a(n), b(n) and DE and provides to second means 19 (a(n)+b(n)+DE)/2. The division by two is a shift of the bit position by one. Second means receives a control signal sign(|a(n)-b(n)|-DE) and accordingly selects whether to provide MAX(a(n),b(n)) provided by second

multiplexer 17 or (a(n)+b(n)+DE)/2, provided by first multiplexer 16. First subtracting unit 12 receives a(n), b(n) and DE and generates sign(b(n)-a(n)-DE). Second subtracting unit 13 receives a(n), b(n) and DE and generates sign(a(n)-b(n)-DE). Both sign signals are provided to first multiplexer 16, that selects the appropriate sign signal according to sign(a(n)-b(n)) provided by third subtracting unit 14.

[0034] Conveniently, apparatus 10 performs said calculation in a single clock cycle.

[0035] Referring to Fig. 3 illustrating an apparatus 20 for implementing a linearly approximated Log MAP algorithm, according to a preferred embodiment of the invention.

[0036] Apparatus 20 implements the following equations:

$$[3] \qquad MAX^*(a(n), b(n)) = a(n)$$

when (a(n)-b(n)-DE) > 0 and (b(n)-a(n) -DE) < 0.

$$[4] \qquad MAX^*(a(n), b(n)) = b(n)$$

when (a(n)-b(n)-DE) < 0 and (b(n)-a(n) -DE) > 0.

$$[5] \qquad MAX^*(a(n), b(n)) = (a(n)+b(n)+DE)/2$$

when |a(n)-b(n)| is smaller than DE.

[0037] Apparatus 20 allows to calculate in parallel the following values: (a(n)+b(n)+DE)/2, the sign of (a(n)-b(n) -DE) and the sign of (b(n)-a(n)-DE) and to determine, according to said values which of following values to provide as a result of the calculation : a(n), b(n) and (a(n)+b(n)+DE)/2.

[0038] Accordingly, apparatus 20 comprises of first means 21 for calculating the following values: (a(n)+b(n)+DE)/2, the sign of (a(n)-b(n) -DE) and sign of (b(n)-a(n)-DE) and a second means 29 to determine which of the following elements to provide (a(n), b(n) or (a(n)+b(n)+DE)/2) as a result, according to said sign values.

[0039] First means 21 comprising of first to second subtracting units 22-23, adder 24. First and second subtracting units 22 - 23 and adder 24 receive a(n), b(n) and DE. Adder 24 receives a(n), b(n) and DE and provides to second means 29 (a(n)+b(n)+DE)/2. The division by two is a shift of the bit position by one. First subtracting unit 22 generates sign(b(n)-a(n)-DE). Second subtracting unit 23 generates sign(a(n)-b(n)-DE). Both sign signals are provided to control inputs of second means 29. Second means 29 is conveniently a multiplexer that receives a(n), b(n) and (a(n)+b(n)+DE)/2 and selects which signal to provide according to said control signals.

[0040] Conveniently, apparatus 20 performs said calculation in a single clock cycle.

[0041] FIG. 4 is a flow chart illustrating a method 30 for implementing a linearly approximated Log MAP algorithm.

[0042] Method 30 starts at step 31 of receiving a(n) and b(n).

[0043] Step 31 is followed by step 32 of calculating (a(n)+b(n)+DE)/2 and determining at least one intermediate result reflecting at least one relationship between a(n), b(n) and DE.

[0044] Step 32 is followed by step 33 of providing a one of the values of a(n), b(n) or (a(n)+b(n)+DE)/2, dependent upon the at least one intermediate result.

[0045] It should be noted that the particular terms and expressions employed and the particular structural and operational details disclosed in the detailed description and accompanying drawings are for illustrative purposes only and are not intended to in any way limit the scope of the invention as described in the appended claims.

[0046] Thus, there has been described herein an embodiment including at least one preferred embodiment of an apparatus and method for implementing a fast Log-MAP (Maximum-a- Posteriori) algorithm for turbo decoding and turbo equalization.

[0047] It will be apparent to those skilled in the art that the disclosed subject matter may be modified in numerous ways and may assume many embodiments other than the preferred form specifically set out and described above.

Claims

1. A method for implementing a linearly approximated Log MAP algorithm involving the calculation of a MAX* (a(n),b(n)) function, said method comprising the steps of:

   receiving a(n), b(n) and a value DE;
   calculating (a(n)+b(n)+DE)/2 and generating at least one intermediate result, said at least one intermediate result reflecting at least one relationship between at least two elements out of a(n), b(n) and DE; and
   providing an MAX*(a(n),b(n)) result selected from a group comprising of a(n), b(n) or (a(n)+b(n)+DE)/2, said selection dependent upon the at least one intermediate result.

2. An iterative decoding method including the linearly approximated Log MAP algorithm implementing method of claim 1.

3. An iterative equalization method including the linearly approximated Log MAP algorithm implementing method of claim 1.

4. The method of claim 1 wherein a first intermediate result indicates whether an absolute value of a difference between a(n) and b(n) is greater than DE.

5. The method of claim 1 a second intermediate result indicates whether a(n) is greater than b(n).

6. The method of claim 5 wherein a first intermediate result indicates whether an absolute value of a difference between a(n) and b(n) is greater than DE.

7. The method of claim 1 wherein the at least one intermediate result comprising of a sign(a(n)-b(n)-DE) signal and a sign(b(n)-a(n)-DE) signal.

8. The method of claim 1 wherein the at least one intermediate result comprising of a sign(a(n)-b(n)) signal and a sign(|a(n)-b(n)|-DE) signal.

9. The method of claim 1 wherein said method is executed during 2 or 3 clock cycles.

10. The method of claim 1 wherein the MAX*(a(n),b(n)) result equals MAX(a(n),b(n)) when |a(n)-b(n)| is not smaller than DE, and equals (a(n)+b(n)+DE)/2 when |a(n)-b(n)| is smaller than DE.

11. The method of claim 1 wherein the MAX*(a(n), b(n)) result equals a(n) when (a(n)-b(n)-DE) > 0 and (b(n)-a(n) -DE) < 0; wherein the MAX*(a(n), b(n)) result equals b(n) when (a(n)-b(n)-DE) < 0 and (b(n)-a(n) -DE) > 0; and
    wherein the MAX*(a(n), b(n)) result equals (a(n) +b (n) +DE)/2 when (a(n)-b(n)-DE) < 0 and (b(n)-a(n)-DE) < 0.

12. An apparatus for implementing a linearly approximated Log MAP algorithm involving the calculation of a MAX* (a(n),b(n)) function, said apparatus comprising:

    calculating means for receiving a(n), b(n) and a value DE;
    means for calculating (a(n)+b(n)+DE)/2 and for generating at least one intermediate result, said at least one intermediate result reflecting at least one relationship between at least two elements out of a(n), b(n) and DE; and
    selecting means for providing an MAX*(a(n),b(n)) result selected from a group comprising of a(n), b(n) or (a(n)+b(n)+DE)/2, said selection dependent upon the at least one intermediate result.

13. An iterative decoder comprising linearly approximated Log MAP algorithm implementing apparatus of claim 12.

14. An iterative equalizer comprising the linearly approximated Log MAP algorithm implementing appa-

ratus of claim 12.

15. The apparatus of claim 12 wherein a first intermediate result indicates whether an absolute value of a difference between a(n) and b(n) is greater than DE.

16. The apparatus of claim 12 a second intermediate result indicates whether a(n) is greater than b(n).

17. The apparatus of claim 16 wherein a first intermediate result indicates whether an absolute value of a difference between a(n) and b(n) is greater than DE.

18. The apparatus of claim 12 wherein the at least one intermediate result comprising of a sign(a(n)-b(n)-DE) signal and a sign(b(n)-a(n)) signal.

19. The apparatus of claim 12 wherein the at least one intermediate result comprising of a sign(a(n)-b(n)) signal and a sign(|a(n)-b(n)|-DE) signal.

20. The apparatus of claim 12 wherein said apparatus calculates the MAX*(a(n),b(n)) function during a single clock cycle.

21. The apparatus of claim 12 wherein the MAX*(a(n), b(n)) result equals MAX(a(n),b(n)) when |a(n)-b(n)| is not smaller than DE, and equals (a(n)+b(n)+DE)/2 when |a(n)-b(n)| is smaller than DE.

22. The apparatus of claim 12 wherein the MAX*(a(n), b(n))result equals a(n) when (a(n)-b(n)-DE) > 0 and (b(n)-a(n) -DE) < 0; wherein the MAX*(a(n), b(n)) result equals b(n) when (a(n)-b(n)-DE) < 0 and (b(n)-a(n) -DE) > 0; and
    wherein the MAX*(a(n), b(n)) result equals (a(n)+b(n)+DE)/2 when (a(n)-b(n)-DE) < 0 and (b(n)-a(n)-DE) < 0.

Patentansprüche

1. Verfahren zum Implementieren einer linearen Approximation des Log-MAP-Algorithmus, das die Berechnung einer MAX*(a(n), b(n))-Funktion einschließt, wobei das Verfahren die folgenden Schritte umfasst:

    Empfangen von a(n), b(n) und eines Wertes DE;
    Berechnen von (a(n)+b(n)+DE)/2 und Erzeugen mindestens eines Zwischenergebnisses, wobei das mindestens eine Zwischenergebnis mindestens eine Beziehung zwischen mindestens zwei Elementen der Elemente a(n), b(n) und DE widerspiegelt; und

Bereitstellen eines MAX*(a(n), b(n))-Ergebnisses, das aus einer Gruppe ausgewählt wird, die a(n), b(n) oder (a(n)+b(n)+DE)/2 umfasst, wobei die Auswahl von dem mindestens einen Zwischenergebnis abhängt.

2. Iteratives Dekodierverfahren, das das Verfahren zum Implementieren der linearen Approximation des Log-MAP-Algorithmus gemäß Anspruch 1 umfasst.

3. Iteratives Entzerrungsverfahren, das das Verfahren zum Implementieren der linearen Approximation des Log-MAP-Algorithmus gemäß Anspruch 1 umfasst.

4. Verfahren gemäß Anspruch 1, bei dem ein erstes Zwischenergebnis anzeigt, ob ein absoluter Wert einer Differenz zwischen a(n) und b(n) größer als DE ist.

5. Verfahren gemäß Anspruch 1, bei dem ein zweites Zwischenergebnis anzeigt, ob a(n) größer als b(n) ist.

6. Verfahren gemäß Anspruch 5, bei dem ein erstes Zwischenergebnis anzeigt, ob ein absoluter Wert einer Differenz zwischen a(n) und b(n) größer als DE ist.

7. Verfahren gemäß Anspruch 1, bei dem das mindestens eine Zwischenergebnis ein sign(a(n)-b(n)-DE)-Signal und ein sign(b(n)-a(n)-DE)-Signal umfasst.

8. Verfahren gemäß Anspruch 1, bei dem das mindestens eine Zwischenergebnis ein sign(a(n)-b(n))-Signal und ein sign(|a(n)-b(n)|-DE)-Signal umfasst.

9. Verfahren gemäß Anspruch 1, bei dem das Verfahren während 2 oder 3 Taktzyklen durchgeführt wird.

10. Verfahren gemäß Anspruch 1, bei dem das MAX*(a(n), b(n))-Ergebnis gleich MAX(a(n), b(n)) ist, wenn |a(n)-b(n)| nicht kleiner als DE ist, und gleich (a(n)+b(n)+DE)/2 ist, wenn |a(n)-b(n)| kleiner als DE ist.

11. Verfahren gemäß Anspruch 1, bei dem das MAX*(a(n), b(n))-Ergebnis gleich a(n) ist, wenn (a(n)-b(n)-DE) > 0 und (b(n)-a(n) -DE) < 0;
    bei dem das MAX*(a(n), b(n))-Ergebnis gleich b(n) ist, wenn (a(n)-b(n)-DE) < 0 und (b(n)-a(n) -DE) > 0; und
    bei dem das MAX*(a(n), b(n))-Ergebnis gleich (a(n)+b(n)+DE)/2 ist, wenn (a(n)-b(n)-DE) < 0 und (b(n)-a(n)-DE) < 0.

12. Vorrichtung zum Implementieren einer linearen Approximation des Log-MAP-Algorithmus, die die Berechnung einer MAX*(a(n), b(n))-Funktion einschließt, wobei die Vorrichtung folgendes umfasst:

> Berechnen von Mitteln zum Empfangen von a(n), b(n) und eines Wertes DE;
> Mittel zum Berechnen von (a(n)+b(n)+DE)/2 und zum Erzeugen von mindestens einem Zwischenergebnis, wobei das mindestens eine Zwischenergebnis mindestens eine Beziehung zwischen mindestens zwei Elementen der Elemente a(n), b(n) und DE widerspiegelt; und Auswählen von Mitteln zum Bereitstellen eines MAX*(a(n), b(n))-Ergebnisses, das aus einer Gruppe ausgewählt wird, die a(n), b(n) oder (a(n)+b(n)+DE)/2 umfasst, wobei die Auswahl von dem mindestens einen Zwischenergebnis abhängt.

13. Iterativer Dekodierer, der die Vorrichtung zum Implementieren der linearen Approximation des Log-MAP-Algorithmus gemäß Anspruch 12 umfasst.

14. Iterativer Entzerrer, der die Vorrichtung zum Implementieren der linearen Approximation des Log-MAP-Algorithmus gemäß Anspruch 12 umfasst.

15. Vorrichtung gemäß Anspruch 12, bei der ein erstes Zwischenergebnis anzeigt, ob ein absoluter Wert einer Differenz zwischen a(n) und b(n) größer als DE ist.

16. Vorrichtung gemäß Anspruch 12, bei der ein zweites Zwischenergebnis anzeigt, ob a(n) größer als b(n) ist.

17. Vorrichtung gemäß Anspruch 16, bei der ein erstes Zwischenergebnis anzeigt, ob ein absoluter Wert einer Differenz zwischen a(n) und b(n) größer als DE ist.

18. Vorrichtung gemäß Anspruch 12, bei der das mindestens eine Zwischenergebnis ein sign(a(n)-b(n)-DE)-Signal und ein sign(b(n)-a(n))-Signal umfasst.

19. Vorrichtung gemäß Anspruch 12, bei der das mindestens eine Zwischenergebnis ein sign(a(n)-b(n))-Signal und ein sign(|a(n)-b(n)|-DE)-Signal umfasst.

20. Vorrichtung gemäß Anspruch 12, bei der die Vorrichtung die MAX*(a(n), b(n))-Funktion während eines einzigen Taktzyklus' berechnet.

21. Vorrichtung gemäß Anspruch 12, bei der das MAX* (a(n), b(n))-Ergebnis gleich MAX(a(n), b(n)) ist, wenn |a(n)-b(n)| nicht kleiner als DE ist, und gleich (a(n)+b(n)+DE)/2 ist, wenn |a(n)-b(n)| kleiner als DE ist.

22. Vorrichtung gemäß Anspruch 12, bei der das MAX* (a(n), b(n))-Ergebnis gleich a(n) ist, wenn (a(n)-b(n) -DE) > 0 und (b(n)-a(n) -DE) < 0;

> bei der das MAX*(a(n), b(n))-Ergebnis gleich b(n) ist, wenn (a(n)-b(n)-DE) < 0 und (b(n)-a(n) -DE) > 0; und
> bei der das MAX*(a(n), b(n))-Ergebnis gleich (a(n)+b(n)+DE)/2 ist, wenn (a(n)-b(n)-DE) < 0 und (b(n)-a(n)-DE) < 0.

Revendications

1. Procédé d'implémentation d'un algorithme Log MAP à approximation linéaire, mettant en jeu le calcul d'une fonction MAX*((a(n),b(n)), ledit procédé comprenant les étapes qui consistent à :

> recevoir a(n), b(n) et une valeur DE ;
>
> calculer (a(n)+b(n)+DE)/2 et générer au moins un résultat intermédiaire, ledit au moins un résultat intermédiaire reflétant au moins une relation entre au moins deux des éléments a(n), b(n) et DE ; et
>
> émettre un résultat MAX*((a(n),b(n)) choisi dans un groupe constitué de a(n), b(n) ou (a(n) +b(n)+DE)/2, ledit choix dépendant dudit au moins un résultat intermédiaire.

2. Procédé de décodage itératif, comprenant l'algorithme Log MAP à approximation linéaire implémentant le procédé de la revendication 1.

3. Procédé d'égalisation itératif, comprenant l'algorithme Log MAP à approximation linéaire implémentant le procédé de la revendication 1.

4. Procédé selon la revendication 1, dans lequel un premier résultat intermédiaire indique si une valeur absolue d'une différence entre a(n) et b(n) est supérieure ou non à DE.

5. Procédé selon la revendication 1, dans lequel un deuxième résultat intermédiaire indique si a(n) est supérieur ou non à b(n).

6. Procédé selon la revendication 5, dans lequel un premier résultat intermédiaire indique si une valeur absolue d'une différence entre a(n) et b(n) est supérieure ou non à DE.

7. Procédé selon la revendication 1, dans lequel l'au moins un résultat intermédiaire est constitué d'un signal sign(a(n)-b(n)-DE) et d'un signal sign(b(n)-a(n)-DE).

8. Procédé selon la revendication 1, dans lequel l'au moins un résultat intermédiaire est constitué d'un signal sign(a(n)-b(n)) et d'un signal sign(|a(n)-b(n)|-DE).

9. Procédé selon la revendication 1, dans lequel le procédé est exécuté pendant 2 ou 3 cycles d'horloge.

10. Procédé selon la revendication 1, dans lequel le résultat de MAX*(a(n),b(n)) est égal à MAX(a(n),b(n)) lorsque |a(n)-b(n)| n'est pas inférieur à DE, et est égal à (a(n)+b(n)+DE)/2 lorsque |a(n)-b(n)| est inférieur à DE.

11. Procédé selon la revendication 1, dans lequel :

    le résultat de MAX*(a(n),b(n)) est égal à a(n) lorsque (a(n)-b(n)-DE) > 0 et (b(n)-a(n)-DE) < 0 ;

    le résultat de MAX*(a(n),b(n)) est égal à b(n) lorsque (a(n)-b(n)-DE) < 0 et (b(n)-a(n)-DE) > 0 ; et

    le résultat de MAX*(a(n),b(n)) est égal à (a(n)+b(n)+DE)/2 lorsque (a(n)-b(n)-DE) < 0 et (b(n)-a(n)-DE) < 0.

12. Appareil pour implémenter un algorithme Log MAP à approximation linéaire, mettant en jeu le calcul d'une fonction MAX*((a(n),b(n)), ledit appareil comprenant:

    des moyens de calcul, pour recevoir a(n), b(n) et une valeur DE ;

    des moyens pour calculer (a(n)+b(n)+DE)/2 et générer au moins un résultat intermédiaire, ledit au moins un résultat intermédiaire reflétant au moins une relation entre au moins deux des éléments a(n), b(n) et DE ; et

    des moyens de sélection, pour émettre un résultat MAX*((a(n),b(n)) choisi dans un groupe constitué de a(n), b(n) ou (a(n)+b(n)+DE)/2, ledit choix dépendant dudit au moins un résultat intermédiaire.

13. Décodeur itératif comprenant l'algorithme Log MAP à approximation linéaire implémentant l'appareil de la revendication 12.

14. Egaliseur itératif comprenant l'algorithme Log MAP à approximation linéaire implémentant l'appareil de la revendication 12.

15. Appareil selon la revendication 12, dans lequel un premier résultat intermédiaire indique si une valeur absolue d'une différence entre a(n) et b(n) est supérieure ou non à DE.

16. Appareil selon la revendication 12, dans lequel un deuxième résultat intermédiaire indique si a(n) est supérieur ou non à b(n).

17. Appareil selon la revendication 16, dans lequel un premier résultat intermédiaire indique si une valeur absolue d'une différence entre a(n) et b(n) est supérieure ou non à DE.

18. Appareil selon la revendication 12, dans lequel l'au moins un résultat intermédiaire est constitué d'un signal sign(a(n)-b(n)-DE) et d'un signal sign(b(n)-a(n)).

19. Appareil selon la revendication 12, dans lequel l'au moins un résultat intermédiaire est constitué d'un signal sign(a(n)-b(n)) et d'un signal sign(|a(n)-b(n)|-DE).

20. Appareil selon la revendication 12, dans lequel ledit appareil calcule la fonction MAX*(a(n),b(n)) pendant un seul cycle d'horloge.

21. Appareil selon la revendication 12, dans lequel le résultat de MAX*(a(n),b(n)) est égal à MAX(a(n),b(n)) lorsque |a(n)-b(n)| n'est pas inférieur à DE, et est égal à (a(n)+b(n)+DE)/2 lorsque |a(n)-b(n)| est inférieur à DE.

22. Appareil selon la revendication 12, dans lequel :

    le résultat de MAX*(a(n),b(n)) est égal à a(n) lorsque (a(n)-b(n)-DE) > 0 et (b(n)-a(n)-DE) < 0 ;

    le résultat de MAX*(a(n),b(n)) est égal à b(n) lorsque (a(n)-b(n)-DE) < 0 et (b(n)-a(n)-DE) > 0 ; et

    le résultat de MAX*(a(n),b(n)) est égal à (a(n)+b(n)+DE)/2 lorsque (a(n)-b(n)-DE) < 0 et (b(n)-a(n)-DE) < 0.
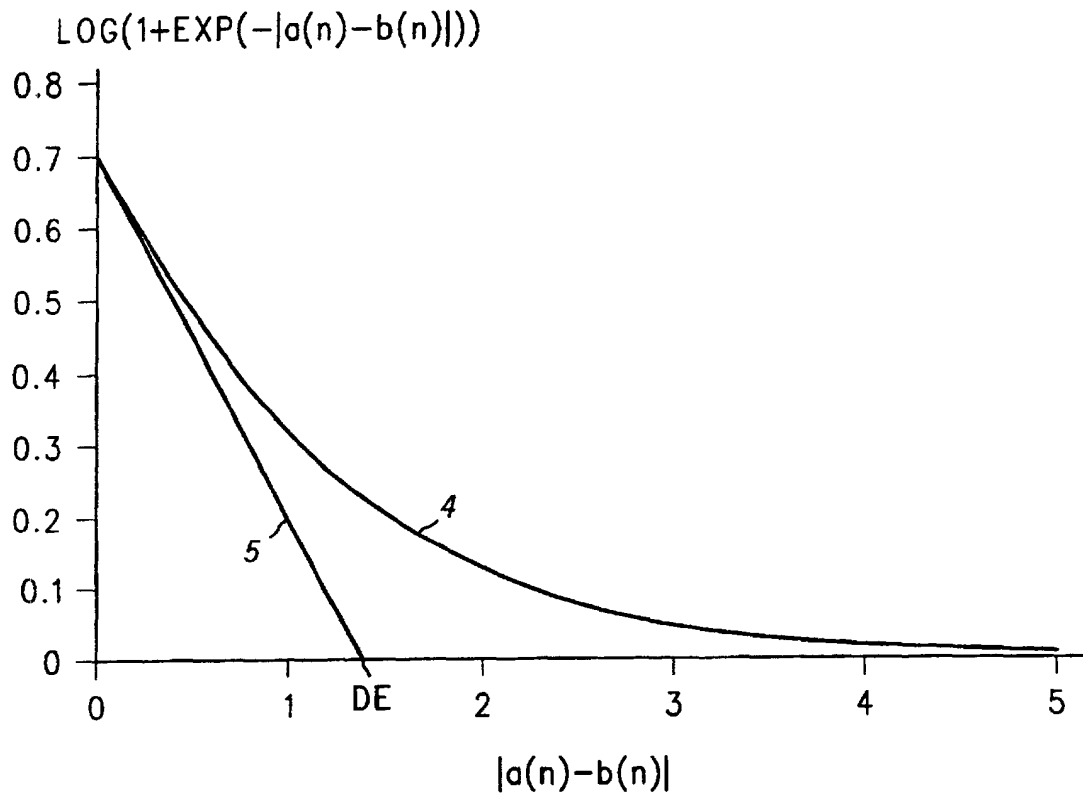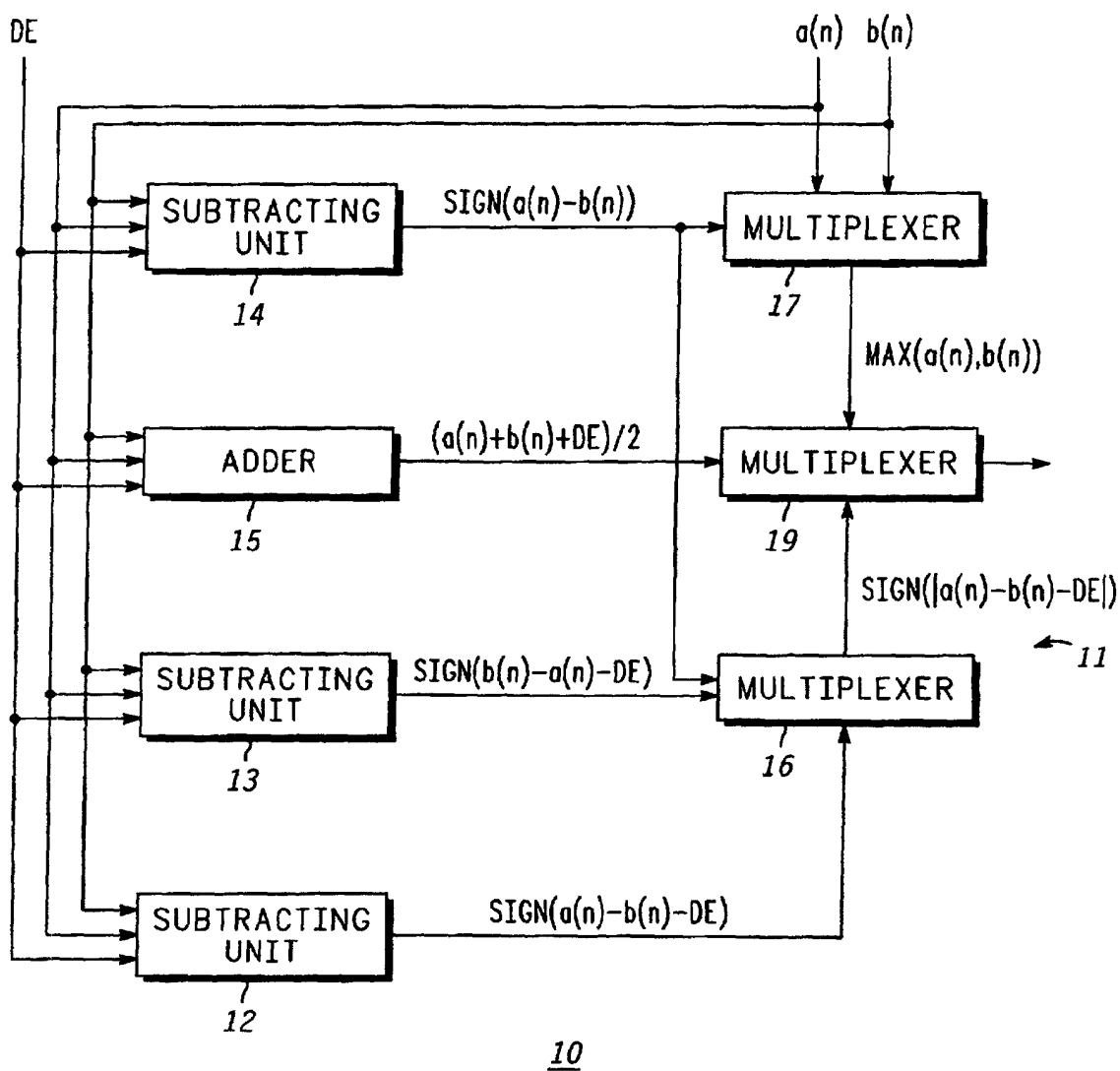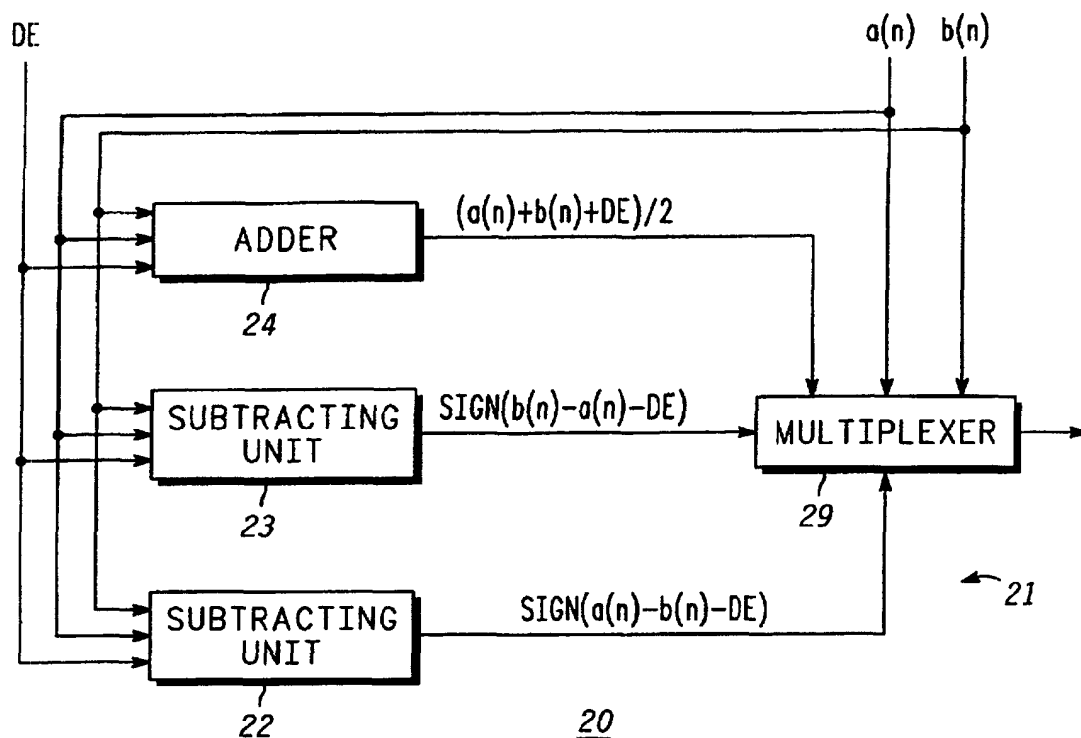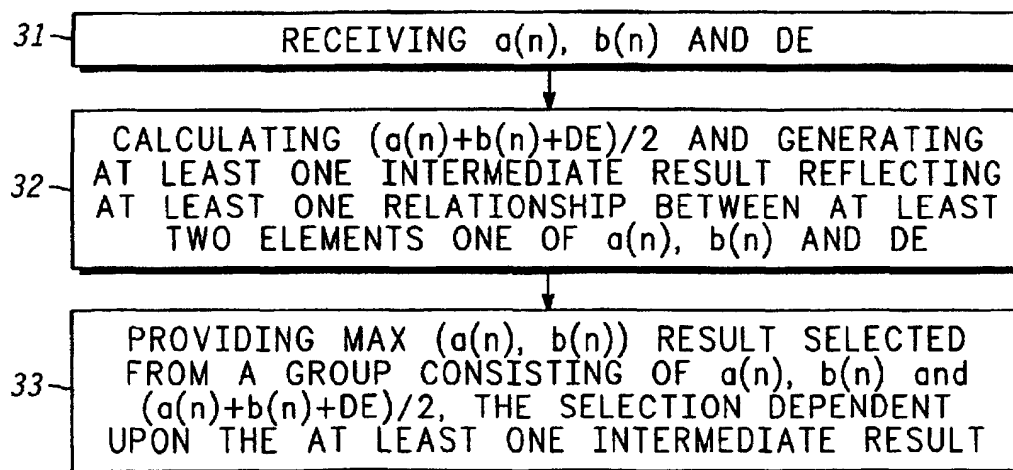
FIG. 1

FIG. 2

DE   a(n) b(n)

```
            ┌──────────────┐   (a(n)+b(n)+DE)/2
            │    ADDER      │
            └──────────────┘
                  24

            ┌──────────────┐   SIGN(b(n)−a(n)−DE)          ┌──────────────┐
            │ SUBTRACTING  │                              │ MULTIPLEXER  │ ──→
            │    UNIT      │                              └──────────────┘
            └──────────────┘                                    29
                  23

            ┌──────────────┐   SIGN(a(n)−b(n)−DE)
            │ SUBTRACTING  │                                           ↖ 21
            │    UNIT      │
            └──────────────┘
                  22                        20
```

FIG. 3

```
31 ─┤  RECEIVING  a(n),  b(n)  AND  DE

32 ─┤  CALCULATING  (a(n)+b(n)+DE)/2  AND  GENERATING
       AT  LEAST  ONE  INTERMEDIATE  RESULT  REFLECTING
       AT  LEAST  ONE  RELATIONSHIP  BETWEEN  AT  LEAST
              TWO  ELEMENTS  ONE  OF  a(n),  b(n)  AND  DE

33 ─┤  PROVIDING  MAX  (a(n),  b(n))  RESULT  SELECTED
       FROM  A  GROUP  CONSISTING  OF  a(n),  b(n)  and
       (a(n)+b(n)+DE)/2,  THE  SELECTION  DEPENDENT
       UPON  THE  AT  LEAST  ONE  INTERMEDIATE  RESULT
```

30

FIG. 4